Atom / VS Code
Using IDEs remotely

Ludwig Böss

18.05.2020

1 Atom

Atom is a free, open source IDE (Integrated Development Environment). You can find information on their website: `https://atom.io/` or the GitHub page: `https://github.com/atom/atom`

1.1 Packages

Atom is very modular and you can add functionality by installing packages. We will only look at a small subset that I find practical.
You can find more packages here: `https://atom.io/packages`
Packages can be installed via the internal package manager, or via console.

1.1.1 Git integration

You can connect to Git repositories and use Atom as a simple interface to stage/stash files, push commits, merge pull requests, etc.
This functionality is built-in!
Help with setup: `https://www.youtube.com/watch?v=6HsZMl-qV5k`

1.1.2 atom-latex

Compiles and shows latex files in Atom. `https://atom.io/packages/atom-latex`

1.1.3 ftp-remote-edit

The workhorse for today. Allows you to add severs and remotely edit files on there. `https://atom.io/packages/ftp-remote-edit`
Problem: Does not (to my knowledge) support daisy-chaining. You can only access servers that are visible to the outside world.

1.1.4 pdf-view

Allows you to render pdf files within Atom. `https://atom.io/packages/pdf-view`

1.1.5 atom-beautify

Auto-format code. `https://atom.io/packages/atom-beautify`
Wrapper library. "Beautifiersfor different languages must be installed individually.

1.1.6 Hydrogen

Use Jupyter Kernels in Atom. `https://atom.io/packages/Hydrogen`
You can either start a local kernel, or connect to a remote kernel in the same way as to a remote notebook (see Til's talk!). Individual lines can be executed and output is shown next to them. Also shows plots.
Help with remote kernel setup: `https://nteract.gitbooks.io/hydrogen/docs/Usage/RemoteKernelConnection.html`

1.1.7 Juno

Julia specific package, similar to Hydrogen, but with more functionality. `https://junolab.org/`
Practical side-bar, integrated plot-pane, workspace to show variables, documentation browser and debugger.

1.1.8 teletype

Provides another person remote access to your editor instance. `https://teletype.atom.io/`
Practical for working simultaniously on the same code, e.g. as a supervisor.

1.1.9 vim-mode-plus

Vim keybindings and shortcuts for Atom. `https://github.com/t9md/atom-vim-mode-plus`

1.1.10 color-picker

Gives you a simple way to pick RGB/hex colors within Atom. `https://atom.io/packages/color-picker`

1.1.11 file-icons

Shows filetype specific icons next to the file trees. `https://atom.io/packages/file-icons`

1.2 Themes

The appearance can be modified on two levels: UI and Syntax highlighting. The list of themes to choose from is extensive, I will only give some examples.
More info can be found here: `https://atom.io/themes`

1.2.1 UI Themes

Default: One Dark/Light
Google style: Atom Material
Sublime Text: Sublime Dark

1.2.2 Syntax Themes

Default: One Dark/Light
Sublime: Monokai Sublime

2 VS Code

VS Code is the free and open source version of Microsofts Visual Studio. According to a 2019 survey on Stack Overflow it is the most frequently used editor among the surveyed programmers.

2.1 Extensions

Similar to Atom VS Code is modular and can be extended. These extensions can be install via the internal marketplace. I will again give a short overview of some useful packages.

2.1.1 Git integration

Allows to execute git commands within VS Code. The design and especially the representation of git diff is slightly better, in my opinion.
Can be extended with GitLens. `https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens`.

2.1.2 Python

Native suport for Python execution, as well as Jupter notebooks.

2.1.3 LaTeX Workshop

Works the same as in Atom. Builds the project on save.
VERY good autocomplete and other cool functionality: shows equation and figure preview on mouse hover.

2.1.4 Remote - SSH

In contrast to ftp-remote-edit this starts a full remote instance of VS Code on the server. That makes it more stable and allows to use all features of VS Code on the remote machine.
In principle you can daisy chain servers by defining a jump-host between the target host (which is not visible to the outside world) and the user pc.
Info:
`https://code.visualstudio.com/docs/remote/ssh`
`https://code.visualstudio.com/blogs/2019/10/03/remote-ssh-tips-and-tricks`

Caution: On Windows you need to patch your SSH install for this to work!
See: `https://github.com/microsoft/vscode-remote-release/issues/18`

Example Config to run on USM machines:

```
Host <jump_host>
  HostName <jump_host>.usm.uni-muenchen.de
  User <user>
  ForwardX11 yes


Host <target_host>
  HostName <target_host>.usm.uni-muenchen.de
  User <user>
  ForwardX11Trusted yes
  AddressFamily inet
  ProxyCommand ssh -W %h:%p <jump_host>
```

2.1.5 Live Share

Allows you to share your editor instance with another user. Similar to the Atom package you share a link and invite someone into your editor instance.
Shares your current workspace and everyone invited can access files. So be aware of that.

2.1.6 VIM

vim functionality and shortcuts. `https://marketplace.visualstudio.com/items?itemName=vscodevim.vim`

2.2 Themes

Again, like Atom, VS Code can be customized and you can change appearance and syntax highlightling themes.
For the UI a popular package seems to be Peacock: `https://marketplace.visualstudio.com/items?itemName=johnpapa.vscode-peacock`
You can install e.g. the Atom syntax theme: `https://marketplace.visualstudio.com/items?itemName=zhuangtongfa.Material-theme`